

04/26/00

1c760 U.S. PTO

0055377 042600

PTO  
09/558377  
04/26/00

A

Please type a plus sign (+) inside this box → ☒Approved for use through 09/30/2000. OMB 0651-0032  
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

|  |  |   |
|--|--|---|
| <b>UTILITY<br/>PATENT APPLICATION<br/>TRANSMITTAL</b><br><small>(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))</small> | Attorney Docket No.                      | T2147-906398  |
|  | First Inventor or Application Identifier | Jean-Francois AUTECHAUD   |
|  | Title                                    | Modular Interconnection Architecture for an Expandable Multiprocessor Machine, Using... |
| Express Mail Label No.   |  |   |

|   |   |  |  |
|---|---|--|--|
| <b>APPLICATION ELEMENTS</b><br><small>See MPEP chapter 600 concerning utility patent application contents.</small>  |   | <b>ADDRESS TO:</b><br>Assistant Commissioner for Patents<br>Box Patent Application<br>Washington, DC 20231   |  |
| 1. <input checked="" type="checkbox"/> * Fee Transmittal Form (e.g., PTO/SB/17)<br><small>(Submit an original and a duplicate for fee processing)</small>   | 5. <input type="checkbox"/> Microfiche Computer Program (Appendix)  | <b>ACCOMPANYING APPLICATION PARTS</b><br>7. <input checked="" type="checkbox"/> Assignment Papers (cover sheet & document(s))<br>8. <input type="checkbox"/> 37 C.F.R. § 3.73(b) Statement of Power of Attorney<br>9. <input checked="" type="checkbox"/> English Translation Document (if applicable) w/ Verification of Translation<br>10. <input checked="" type="checkbox"/> Information Disclosure Statement (IDS)/PTO-1449 <input checked="" type="checkbox"/> Copies of IDS Citations<br>11. <input checked="" type="checkbox"/> Preliminary Amendment<br>12. <input checked="" type="checkbox"/> Return Receipt Postcard (MPEP 503) (Should be specifically itemized)<br>13. <input type="checkbox"/> * Small Entity Statement(s) filed in prior application, Status still proper and desired (PTO/SB/09-12)<br>14. <input checked="" type="checkbox"/> Certified Copy of Priority Document(s) (if foreign priority is claimed) FR99/05266<br>15. <input checked="" type="checkbox"/> Other Proposed Drawing Corrections |  |
| 2. <input checked="" type="checkbox"/> Specification [Total Pages 21]<br><small>(preferred arrangement set forth below)</small><br>- Descriptive title of the Invention<br>- Cross References to Related Applications<br>- Statement Regarding Fed sponsored R & D<br>- Reference to Microfiche Appendix<br>- Background of the Invention<br>- Brief Summary of the Invention<br>- Brief Description of the Drawings (if filed)<br>- Detailed Description<br>- Claim(s)<br>- Abstract of the Disclosure | 6. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)<br>a. <input type="checkbox"/> Computer Readable Copy<br>b. <input type="checkbox"/> Paper Copy (identical to computer copy)<br>c. <input type="checkbox"/> Statement verifying identity of above copies |  |  |
| 3. <input checked="" type="checkbox"/> Drawing(s) (35 U.S.C. 113) [Total Sheets 5]<br>formal  |   |  |  |
| 4. Oath or Declaration [Total Pages C]<br>a. <input checked="" type="checkbox"/> Newly executed (original or copy)<br>b. <input type="checkbox"/> Copy from a prior application (37 C.F.R. § 1.63(d)) (for continuation/divisional with Box 16 completed)<br>i. <input type="checkbox"/> DELETION OF INVENTOR(S)<br>Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).   |   |  |  |
| <b>NOTE FOR ITEMS 1 &amp; 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).</b>  |   |  |  |

16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: \_\_\_\_\_

Prior application information: Examiner \_\_\_\_\_ Group / Art Unit: \_\_\_\_\_

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

## 17. CORRESPONDENCE ADDRESS

☐ Customer Number or Bar Code Label \_\_\_\_\_ or ☒ Correspondence address below  
 (Insert Customer No. or Attach bar code label here)

|         |                                 |           |              |          |              |
|---------|---------------------------------|-----------|--------------|----------|--------------|
| Name    | Edward J. Kondracki             |           |              |          |              |
|         | MILES & STOCKBRIDGE P.C.        |           |              |          |              |
| Address | 1751 Pinnacle Drive - Suite 500 |           |              |          |              |
| City    | McLean                          | State     | VA           | Zip Code | 22102-3833   |
| Country | U.S.                            | Telephone | 703/903-9000 | Fax      | 703/610-8686 |

|                   |                            |                                   |                |
|-------------------|----------------------------|-----------------------------------|----------------|
| Name (Print/Type) | Edward J. Kondracki        | Registration No. (Attorney/Agent) | 20,604         |
| Signature         | <i>Edward J. Kondracki</i> | Date                              | April 26, 2000 |

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of: :  
Jean-Francois AUTECHAUD : Examiner:  
Serial No.: To Be Assigned : Group Art Unit:  
Filed: April 26, 2000 : Corres. To FR99/05266  
For: MODULAR INTERCONNECTION : Filed April 26, 1999  
ARCHITECTURE FOR AN EXPANDABLE :  
MULTIPROCESSOR MACHINE, USING A :  
MULTILEVEL BUS HIERARCHY AND THE :  
SAME BUILDING BLOCK FOR ALL THE :  
LEVELS :

McLean, Virginia

**PRELIMINARY AMENDMENT**

Honorable Commissioner of Patents and Trademarks  
Washington, DC 20231

Sir:

Prior to examination of the above-identified application, please amend the application as follows:

**IN THE SPECIFICATION:**

Page 1, after the title and before the first paragraph, insert the following headings:

**--BACKGROUND OF THE INVENTION**

**Field of the Invention--;**

Page 1, line 12, before the paragraph beginning "In such machines...", insert the following heading at the left hand margin:

**--Description of Related Art--**

Page 4, line 9, before the paragraph beginning "The object of the invention...", insert the following heading at the left hand margin:

--Summary of the Invention--

Page 5, at line 28, before the paragraph beginning "Other advantages...", insert the following heading at the left hand margin:

--Brief Description of the Drawings--;

Page 6, line 17, before the paragraph beginning "Fig. 1...", insert the following heading at the left hand margin:

--Description of the Preferred Embodiment(s) --;

Page 18, after the last paragraph ending "...quasi-UMA machine.", insert the following new paragraph:

--While this invention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the preferred embodiments of the invention as set forth herein, are intended to be illustrative, not limiting. Various changes may be made without departing from the true spirit and full scope of the invention as set forth herein and defined in the claims—

**IN THE CLAIMS:**

Please cancel claims 1-12 in their entirety and without prejudice and add the following new claims:

- 1           --13. A modular interconnection architecture for an expandable
- 2 multiprocessor machine, based on a virtual bus hierarchy, comprising a given number
- 3 of multiprocessor modules (QPi) , each module including a plurality of processors and
- 4 associated cache memories organized into nodes (Nj) and distributed on at least two
- 5 interconnection levels: a first interconnection level (MI) corresponding to
- 6 interconnection of the multiprocessor modules (QPi) within a node (Nj), and a second
- 7 interconnection level (SI) corresponding to the interconnection of the nodes (Nj) with
- 8 one another, the first interconnection level (MI) comprising connection agents (NCSi)
- 9 connecting the multiprocessor modules (QPi) to one another and handling the

transactions between the multiprocessor modules (QPi), the second interconnection level (SI) comprising external connection nodes (NCEj) connecting the nodes (Nj) to one another and handling the transactions between the nodes (Nj), the connection agents (NCSi) and the external connection nodes (NCEj) respectively having the same basic structure, the same external interface (XI), and adapted to implement the same coherency control protocol for the cache memories of the processors.

14. A modular interconnection architecture according to claim 13, characterized in that each external connection node (NCEj) comprises two identical connection agents (NCSi) connected head-to-tail, one of the two agents (NCS'j) receiving and filtering transactions sent by the node (Nj) to which it is connected, and the other agent (NCS''j) receiving and filtering the transactions sent by the other nodes (Nj) to which it is connected.

15. A modular interconnection architecture according to claim 13, characterized in that each connection agent (NCSi) comprises an associative memory (DDi) with a fixed size determined as a function of the number of processors in the multiprocessor module (QPi) to which the connection agent (NCSi) is connected, the state of the memories (DDi) being indicative of the presence of the last modified data blocks in the cache memories of the multiprocessor module (QPi).

16. A modular interconnection architecture according to claim 14, characterized in that each connection agent (NCSi) comprises an associative memory (DDi) with a fixed size determined as a function of the number of processors in the multiprocessor module (QPi) to which the connection agent (NCSi) is connected, the state of the memories (DDi) being indicative of the presence of the last modified data blocks in the cache memories of the multiprocessor module (QPi).

17. A modular interconnection architecture according to claim 14, characterized in that the first and second head-to-tail connection agents (NCS'j and NCS''j) only accept transactions for blocks modified in their respective associative memories (DD'j and DD''j); modified data blocks in the first connection agent (NCS'j) being exported to the requesting multiprocessor module or modules and, conversely,

6 modified data blocks in the second connection agent (NCS"j) being imported from the  
7 module or modules holding the blocks.

1 18. A modular interconnection architecture according to claim 13,  
2 characterized in that the second interconnection level (SI) has a latency that is double  
3 the latency of the first interconnection level (MI).

1 19. A modular interconnection architecture according to claim 14,  
2 characterized in that the second interconnection level (SI) has a latency that is double  
3 the latency of the first interconnection level (MI).

1 20. A modular interconnection architecture according to claim 15,  
2 characterized in that the second interconnection level (SI) has a latency that is double  
3 the latency of the first interconnection level (MI).

1 21. A modular interconnection architecture according to claim 16,  
2 characterized in that the second interconnection level (SI) has a latency that is double  
3 the latency of the first interconnection level (MI).

1 22. A modular interconnection architecture according to claim 17,  
2 characterized in that the second interconnection level (SI) has a latency that is double  
3 the latency of the first interconnection level (MI).

1 23. A process for expanding the capacity of a machine comprising a first  
2 given number of processors on a first level (MI) organized into a first given number  
3 of multiprocessor modules (QPi) and capable of being inserted into an interconnection  
4 architecture comprising a modular interconnection architecture for an expandable  
5 multiprocessor machine, based on a virtual bus hierarchy, comprising a given number  
6 of multiprocessor modules (QPi), each comprising a plurality of processors and  
7 associated cache memories organized into nodes (Nj) and distributed on at least two  
8 interconnection levels: a first interconnection level (MI) corresponding to  
9 interconnection of the multiprocessor modules (QPi) within a node (Nj), and a second  
10 interconnection level (SI) corresponding to the interconnection of the nodes (Nj) with

one another, the first interconnection level (MI) comprising connection agents (NCSi) connecting the multiprocessor modules (QPi) to one another and handling the transactions between the multiprocessor modules (QPi), the second interconnection level (SI) comprising external connection nodes (NCEj) connecting the nodes (Nj) to one another and handling the transactions between the nodes (Nj), the connection agents (NCSi) and the external connection nodes (NCEj) respectively having the same basic structure, the same external interface (XI), and adapted to implement the same coherency control protocol for the cache memories of the processors, characterized in that it consists of disconnecting one of the first-level multiprocessor modules (QPi) from its connection agent (NCSi) to free said connecting agent and of connecting, via said freed connection agent, a second given number of processors organized into a second given number of multiprocessor modules, also capable of being inserted into said interconnection architecture.

24. A process for expanding the capacity of a machine comprising a first given number of processors on a first level (MI) organized into a first given number of multiprocessor modules (QPi) and capable of being inserted into an interconnection architecture comprising a modular interconnection architecture for an expandable multiprocessor machine, based on a virtual bus hierarchy, comprising a given number of multiprocessor modules (QPi), each comprising a plurality of processors and associated cache memories organized into nodes (Nj) and distributed on at least two interconnection levels: a first interconnection level (MI) corresponding to interconnection of the multiprocessor modules (QPi) within a node (Nj), and a second interconnection level (SI) corresponding to the interconnection of the nodes (Nj) with one another, the first interconnection level (MI) comprising connection agents (NCSi) connecting the multiprocessor modules (QPi) to one another and handling the transactions between the multiprocessor modules (QPi), the second interconnection level (SI) comprising external connection nodes (NCEj) connecting the nodes (Nj) to one another and handling the transactions between the nodes (Nj), the connection agents (NCSi) and the external connection nodes (NCEj) respectively having the same basic structure, the same external interface (XI), and adapted to implement the same coherency control protocol for the cache memories of the processors, each external connection node (NCEj) comprises two identical connection agents (NCSi) connected

head-to-tail, one of the two agents (NCS'<sub>j</sub>) receiving and filtering transactions sent by the node (N<sub>j</sub>) to which it is connected, and the other (NCS''<sub>j</sub>) receiving and filtering the transactions sent by the other nodes (N<sub>j</sub>) to which it is connected, characterized in that it consists of disconnecting one of the first-level multiprocessor modules (QP<sub>i</sub>) from its connection agent (NCS<sub>i</sub>) to free said connecting agent and of connecting, via said freed connection agent, a second given number of processors organized into a second given number of multiprocessor modules, also capable of being inserted into said interconnection architecture.

25. A process for expanding the capacity of a machine comprising a first given number of processors on a first level (MI) organized into a first given number of multiprocessor modules (QP<sub>i</sub>) and capable of being inserted into an interconnection architecture comprising a modular interconnection architecture for an expandable multiprocessor machine, based on a virtual bus hierarchy, comprising a given number of multiprocessor modules (QP<sub>i</sub>), each comprising a plurality of processors and associated cache memories organized into nodes (N<sub>j</sub>) and distributed on at least two interconnection levels: a first interconnection level (MI) corresponding to interconnection of the multiprocessor modules (QP<sub>i</sub>) within a node (N<sub>j</sub>), and a second interconnection level (SI) corresponding to the interconnection of the nodes (N<sub>j</sub>) with one another, the first interconnection level (MI) comprising connection agents (NCS<sub>i</sub>) connecting the multiprocessor modules (QP<sub>i</sub>) to one another and handling the transactions between the multiprocessor modules (QP<sub>i</sub>), the second interconnection level (SI) comprising external connection nodes (NCE<sub>j</sub>) connecting the nodes (N<sub>j</sub>) to one another and handling the transactions between the nodes (N<sub>j</sub>), the connection agents (NCS<sub>i</sub>) and the external connection nodes (NCE<sub>j</sub>) respectively having the same basic structure, the same external interface (XI), and adapted to implement the same coherency control protocol for the cache memories of the processors, characterized in that it consists of disconnecting one of the first-level multiprocessor modules (QP<sub>i</sub>) from its connection agent (NCS<sub>i</sub>) to free said connecting agent and of connecting, via said freed connection agent, a second given number of processors organized into a second given number of multiprocessor modules, also capable of being inserted into said interconnection architecture, each connection agent (NCS<sub>i</sub>) comprises an associative memory (DD<sub>i</sub>) with a fixed size determined as a function of the number of

25 processors in the multiprocessor module (QPi) to which the connection agent (NCSi)  
 26 is connected, the state of the memories (DDi) being indicative of the presence of the  
 27 last modified data blocks in the cache memories of the multiprocessor module (QPi).

1 26. A process according to claim 24, characterized in the connection agents  
 2 (NCS'j) and (NCS''j) respectively comprise an associative memory (DD'j) and  
 3 (DD''j) with a fixed size determined as a function of the number of processors in the  
 4 multiprocessor module (HBj) to which the connection agents (NCS'j) and (NCS''j)  
 5 are connected, the state of the associated memories (DD'j) and (DD''j) being  
 6 indicative of the presence of the last modified data blocks executed or conversely  
 7 imported.

1 27. A process for expanding the capacity of a machine according to claim 24  
 2 wherein the first and second head-to-tail connection agents (NCS'j and NCS''j) only  
 3 accept transactions for blocks modified in their respective associative memories (DD'j  
 4 and DD''j); modified data blocks in the first connection agent (NCS'j) being exported  
 5 to the requesting multiprocessor module or modules and, conversely, modified data  
 6 blocks in the second connection agent (NCS''j) being imported from the module or  
 7 modules holding the blocks, characterized in that it consists of disconnecting one of  
 8 the first-level multiprocessor modules (QPi) from its connection agent (NCSi) to free  
 9 said connecting agent and of connecting, via this freed connection agent, a second  
 10 given number of processors organized into a second given number of multiprocessor  
 11 modules, also capable of being inserted into said interconnection architecture  
 12 comprising a modular interconnection architecture.

1 28. A process for expanding the capacity of a machine according to claim  
 2 24 wherein the second interconnection level (SI) has a latency that is double the  
 3 latency of the first interconnection level (MI), characterized in that it consists of  
 4 disconnecting one of the first-level multiprocessor modules (QPi) from its connection  
 5 agent (NCSi) to free said connecting agent and of connecting, via this freed  
 6 connection agent, a second given number of processors organized into a second given  
 7 number of multiprocessor modules, also capable of being inserted into said  
 8 interconnection architecture.



1           29.     A process according to claim 23, characterized in that, the second  
2     given number of processors being organized into a second given number of  
3     multiprocessor modules on a second level (SI), it consists of connecting it to the  
4     connection agent (NCS<sub>i</sub>) of the first given number of processors on the first level (MI)  
5     through one of the connection agents (NCS"<sub>j</sub>) on the second level.

1           30.     A process according to claim 29, characterized in that, the second  
2     given number of processors also being on the first level (MI), and further comprising  
3     connecting the respective connection agents (NCS') to the first and second given  
4     numbers of processors, the second level (SI) being reduced to a single link.

1           31.     An expandable multi node multiprocessor machine, comprising an  
2     interconnection architecture including a given number of multiprocessor modules  
3     including a modular interconnection architecture for an expandable multiprocessor  
4     machine, based on a virtual bus hierarchy, comprising a given number of  
5     multiprocessor modules (QPi) , each module including a plurality of processors and  
6     associated cache memories organized into nodes (N<sub>j</sub>) and distributed on at least two  
7     interconnection levels: a first interconnection level (MI) corresponding to  
8     interconnection of the multiprocessor modules (QPi) within a node (N<sub>j</sub>), and a second  
9     interconnection level (SI) corresponding to the interconnection of the nodes (N<sub>j</sub>) with  
10    one another, the first interconnection level (MI) comprising connection agents (NCS<sub>i</sub>)  
11    connecting the multiprocessor modules (QPi) to one another and handling the  
12    transactions between the multiprocessor modules (QPi), the second interconnection  
13    level (SI) comprising external connection nodes (NCE<sub>j</sub>) connecting the nodes (N<sub>j</sub>) to  
14    one another and handling the transactions between the nodes (N<sub>j</sub>), the connection  
15    agents (NCS<sub>i</sub>) and the external connection nodes (NCE<sub>j</sub>) respectively having the same  
16    basic structure, the same external interface (XI), and adapted to implement the same  
17    coherency control protocol for the cache memories of the processors.

1           32.     An expandable multinode multiprocessor machine as set forth in claim  
2     31, characterized in that each external connection node (NCE<sub>j</sub>) comprises two  
3     identical connection agents (NCS<sub>i</sub>) connected head-to-tail, one of the two agents  
4     (NCS'<sub>j</sub>) receiving and filtering transactions sent by the node (N<sub>j</sub>) to which it is

5 connected, and the other agent (NCS"j) receiving and filtering the transactions sent by  
6 the other nodes (Nj) to which it is connected.

1 33. An expandable multinode multiprocessor machine as set forth in claim  
2 31, characterized in that each connection agent (NCSi) comprises an associative  
3 memory (DDi) with a fixed size determined as a function of the number of processors  
4 in the multiprocessor module (QPi) to which the connection agent (NCSi) is  
5 connected, the state of the memories (DDi) being indicative of the presence of the last  
6 modified data blocks in the cache memories of the multiprocessor module (QPi).

1 34. An expandable multinode multiprocessor machine as set forth in claim  
2 31, characterized in that each connection agent (NCSi) comprises an associative  
3 memory (DDi) with a fixed size determined as a function of the number of processors  
4 in the multiprocessor module (QPi) to which the connection agent (NCSi) is  
5 connected, the state of the memories (DDi) being indicative of the presence of the last  
6 modified data blocks in the cache memories of the multiprocessor module (QPi).

1 35. An expandable multinode multiprocessor machine as set forth in claim  
2 31, characterized in that the first and second head-to-tail connection agents (NCS'j  
3 and NCS"j) only accept transactions for blocks modified in their respective  
4 associative memories (DD'j and DD"j); modified data blocks in the first connection  
5 agent (NCS'j) being exported to the requesting multiprocessor module or modules  
6 and, conversely, modified data blocks in the second connection agent (NCS"j) being  
7 imported from the module or modules holding the blocks.

1 36. An expandable multinode multiprocessor machine as set forth in claim  
2 31, characterized in that the second interconnection level (SI) has a latency that is  
3 double the latency of the first interconnection level (MI).

1 37 A process for tracing data blocks in an interconnection architecture for  
2 an expandable microprocessor machine, based on a virtual bus hierarchy, comprising  
3 a given number of multiprocessor modules (QPi) ,each module including a plurality  
4 of processors and associated cache memories organized into nodes (Nj) and

distributed on at least two interconnection levels: a first interconnection level (MI) corresponding to interconnection of the multiprocessor modules (QPi) within a node (Nj), and a second interconnection level (SI) corresponding to the interconnection of the nodes (Nj) with one another, the first interconnection level (MI) comprising connection agents (NCSi) connecting the multiprocessor modules (QPi) to one another and handling the transactions between the multiprocessor modules (QPi), the second interconnection level (SI) comprising external connection nodes (NCEj) connecting the nodes (Nj) to one another and handling the transactions between the nodes (Nj), the connection agents (NCSi) and the external connection nodes (NCEj) respectively having the same basic structure, the same external interface (XI), and adapted to implement the same coherency control protocol for the cache memories of the processors, comprising duplicating on the first level in the associative memories (DDi) only modified data blocks in the cache memories of the multiprocessor modules (QPi) and tracing only the modified blocks inside the node (Nj).

38. A process for tracing data blocks as set forth in claim 37 wherein each external connection node (NCEj) comprises two identical connection agents (NCSi) connected head-to-tail, one of the two agents (NCS'j) receiving and filtering transactions sent by the node (Nj) to which it is connected, and the other agent (NCS''j) receiving and filtering the transactions sent by the other nodes (Nj) to which it is connected.

39. A process for tracing data blocks as set forth in claim 37 wherein each connection agent (NCSi) comprises an associative memory (DDi) with a fixed size determined as a function of the number of processors in the multiprocessor module (QPi) to which the connection agent (NCSi) is connected, the state of the memories (DDi) being indicative of the presence of the last modified data blocks in the cache memories of the multiprocessor module (QPi).

40. A process for tracing data blocks as set forth in claim 38, characterized in that the first and second head-to-tail connection agents (NCS'j and NCS''j) only accept transactions for blocks modified in their respective associative memories (DD'j and DD''j); modified data blocks in the first connection agent (NCS'j) being exported

5 to the requesting multiprocessor module or modules and, conversely, modified data  
 6 blocks in the second connection agent (NCS"j) being imported from the module or  
 7 modules holding the blocks

1 41. A process for tracing data blocks as set forth in claim 38, characterized  
 2 in that the first and second head-to-tail connection agents (NCS'j and NCS"j) only  
 3 accept transactions for blocks modified in their respective associative memories (DD'j  
 4 and DD"j); modified data blocks in the first connection agent (NCS'j) being exported  
 5 to the requesting multiprocessor module or modules and, conversely, modified data  
 6 blocks in the second connection agent (NCS"j) being imported from the module or  
 7 modules holding the blocks.

1 42. A process for tracing data blocks as set forth in claim 13, characterized  
 2 in that the second interconnection level (SI) has a latency that is double the latency of  
 3 the first interconnection level (MI).

1 43. A process for tracing data blocks in a modular interconnection  
 2 architecture for an expandable multiprocessor machine, based on a virtual bus  
 3 hierarchy, comprising a given number of multiprocessor modules (QPi) , each module  
 4 including a plurality of processors and associated cache memories organized into  
 5 nodes (Nj) and distributed on at least two interconnection levels: a first  
 6 interconnection level (MI) corresponding to interconnection of the multiprocessor  
 7 modules (QPi) within a node (Nj), and a second interconnection level (SI)  
 8 corresponding to the interconnection of the nodes (Nj) with one another, the first  
 9 interconnection level (MI) comprising connection agents (NCSi) connecting the  
 10 multiprocessor modules (QPi) to one another and handling the transactions between  
 11 the multiprocessor modules (QPi), the second interconnection level (SI) comprising  
 12 external connection nodes (NCEj) connecting the nodes (Nj) to one another and  
 13 handling the transactions between the nodes (Nj), the connection agents (NCSi) and  
 14 the external connection nodes (NCEj) respectively having the same basic structure,  
 15 the same external interface (XI), and adapted to implement the same coherency  
 16 control protocol for the cache memories of the processors, characterized in that it  
 17 consists, on the second level (SI), of duplicating in the associative memories (DD'j

18 and DD"J) of the connection agents (NCS' and NCS"j) of each external connection  
 19 node (NCEj) only the modified blocks exported, or conversely imported, and of  
 20 tracing only the modified blocks exported, or conversely imported, between each  
 21 node (Nj) of the machine.

1 44. A process for tracing data blocks as set forth in claim 43, wherein each  
 2 external connection node (NCEj) comprises two identical connection agents (NCSi)  
 3 connected head-to-tail, one of the two agents (NCS'j) receiving and filtering  
 4 transactions sent by the node (Nj) to which it is connected, and the other agent  
 5 (NCS"j) receiving and filtering the transactions sent by the other nodes (Nj) to which  
 6 it is connected.

1 45. A process for tracing data blocks as set forth in claim 43, wherein each  
 2 connection agent (NCSi) comprises an associative memory (DDi) with a fixed size  
 3 determined as a function of the number of processors in the multiprocessor module  
 4 (QPi) to which the connection agent (NCSi) is connected, the state of the memories  
 5 (DDi) being indicative of the presence of the last modified data blocks in the cache  
 6 memories of the multiprocessor module (QPi).

1 46. A process for tracing data blocks as set forth in claim 44, wherein each  
 2 connection agent (NCSi) comprises an associative memory (DDi) with a fixed size  
 3 determined as a function of the number of processors in the multiprocessor module  
 4 (QPi) to which the connection agent (NCSi) is connected, the state of the memories  
 5 (DDi) being indicative of the presence of the last modified data blocks in the cache  
 6 memories of the multiprocessor module (QPi).

1 47. A process for tracing data blocks as set forth in claim 44, wherein the  
 2 first and second head-to-tail connection agents (NCS'j and NCS"j) only accept  
 3 transactions for blocks modified in their respective associative memories (DD'j and  
 4 DD"j); modified data blocks in the first connection agent (NCS'j) being exported to  
 5 the requesting multiprocessor module or modules and, conversely, modified data  
 6 blocks in the second connection agent (NCS"j) being imported from the module or  
 7 modules holding the blocks.



Please delete the Abstract at page 11 in its entirety and substitute the following new Abstract:

## --ABSTRACT

The present invention relates to a modular interconnection architecture for an expandable multiprocessor machine. It comprises a first interconnection level (MI) comprising connection agents (NCSi) that connect the multiprocessor modules and handle the transactions between the multiprocessor modules, and a second interconnection level (SI) comprising external connection nodes (NCEj) that connect the nodes (Nj) to one another and handle the transactions between the nodes (Nj). Each external connection node (NCEj) comprises two connection agents identical to the connection agent (NCSi), connected head-to-tail, one of the two agents (NCS'j) receives and filters the transactions sent by the node (Nj) to which it is connected. The other agent (NCS''j) receives and filters the transactions sent by the other nodes (Nj) to which it is connected. Its applications specifically include the construction of an entire range of machines: UMA, QUASI-UMA, NUMA, cluster, etc.--



**REMARKS**

This Preliminary Amendment is made to eliminate informalities in the specification, claims and abstract resulting from a literal translation of the French text, to eliminate the use of multiple dependent claims, and to insert headings to conform the application to U.S. practice.

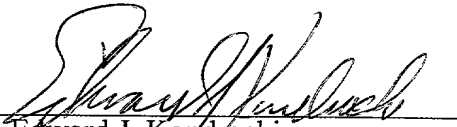
The present application is believed to be in condition for examination, which action is earnestly solicited.

Respectfully submitted,

Miles & Stockbridge P.C.

Date: April 26, 2000

By:

  
Edward J. Kondracki  
Registration No. 20,604

1751 Pinnacle Drive, Suite 500  
McLean, Virginia 22102-3833  
Tel.: (703) 903-9000

### Verification of Translation

I, Robin Holding, having an office at 948 15th Street, #4, Santa Monica, CA 90403-3134, hereby state that I am well acquainted with both the English and French languages and that to the best of my knowledge and ability, the appended document is a true and faithful translation of

**French Patent Application No. 99 05266, filed on April 26, 1999 in the name of BULL S.A., invented by Jean-François AUTECHAUD.**

I further declare that the above statement is true; and further, that this statement is made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent resulting therefrom.

March 31, 2000

Date

Robin Holding  
Robin Holding

**MODULAR INTERCONNECTION ARCHITECTURE FOR AN EXPANDABLE  
MULTIPROCESSOR MACHINE, USING A MULTILEVEL BUS HIERARCHY AND  
THE SAME BUILDING BLOCK FOR ALL THE LEVELS**

5           The present invention relates to the field of multiprocessor machines with symmetrical architectures comprising memories with uniform access and cache coherency, better known by the term "CC-UMA SMP" (Cache Coherent-Uniform Memory Access Symmetrical Multiprocessor).

10           More specifically, it concerns an interconnection architecture between a large number of processors organized into clusters of multiprocessor modules, also called nodes of a computer system.

15           In such machines, one or more levels of cache memory of each processor store data that has been recently acquired and can be reused quickly, thus preventing subsequent contentions for memory access.

20           When a processor does not find the desired data in its own cache memory, it transmits the address of the storage area in which it wishes to read the data via the system bus.

          All the processors check their cache memories to see if they have the most recently updated copy of the data at this address.

25           If another processor has modified the data, it informs the sending processor that it can retrieve the data contained in its cache memory instead of requesting it from physical storage.

          In order for a processor to be able to modify a storage area, it must obtain authorization to do so.

          After such a modification, all the other processors that have a copy of the data that has just been modified must invalidate this copy in their respective cache memories.

30           A trace of the transactions passing through the system bus and a cache coherency control protocol will keep a record of all the transactions exchanged between the processors.

          Conventionally, four fundamental states of the coherency protocol of the memories are distinguished, currently standardized under the code MESI, the abbreviation for "Modified state, Exclusive state, Shared state, Invalid state."

Generally, managing exchanges between the memories of a computer system consists of updating or invalidating copies of a storage block.

If invalidation is chosen, it is known, for example, to add to the address of each block two bits representing the state of the block in the memory.

5 According to this protocol, and in each node, the state "M" corresponds to the case in which a given cache memory is the only holder of a modified copy of a block, called the reference copy, and in which there are no other valid copies in the system. In this case, the cache memory holding the copy having the state "M" is responsible for supplying and updating the internal memory if it wants to dump this copy. A copy having the exclusive state "E" is also a  
10 reference copy held by a cache memory, but in this case, the internal memory has an updated copy of it. The shared state "S" relates to a copy that may have several holders, but only one copy of it is up-to-date in the internal memory. The potential holders depend on the management strategy adopted. Finally, there is the invalid copy state "I" when the cache memory has a copy of the block that is not valid. When a processor wants to access a block in the "I" state, its cache  
15 memory sends a request through the system bus to acquire the block. The other cache memories and the internal memory connected to the system bus learn of this request and react as a function of the state of the copy that the corresponding memory has. Thus, if a cache memory has the block in the modified state "M," it supplies the block to the system bus, thus allowing the processor to request access to the data and its cache memory to acquire the reference copy. There  
20 are still other optional states, which may or may not be standardized.

Management of this type is adopted particularly when several processors frequently modify the data sent via the system bus.

Since the storage areas are all accessible in the same way by all the processors, the problem of optimizing the location of the data in a given storage area no longer exists.

25 That is what characterizes uniform memory access UMA.

This type of management is reliable as long as the system bus is not saturated by too many processors sending too many memory access requests at the same time.

With the increase in the level of component integration, it is possible to store more and more processors on the same card, and processor speed increases by 55% per year.

An article by Alan Charlesworth (Sun Microsystems) published in IEEE Micro, January/February 1998, pages 39-49, describes an architecture for interconnection between several nodes, adapted to such constraints.

This interconnection architecture uses a packet switching protocol that makes it possible to separate the requests and the responses, allowing transactions between several processors to overlap in the bus.

Furthermore, it uses an interleaving of several control busses. Thus, the utilization of four address busses makes it possible to check four addresses in parallel. The physical storage space is divided by four and each address bus checks a quarter of the memory.

With such an architecture, it is possible to connect between 24 to 64 processors using 4 address busses and an automatic switching device, also known as a "crossbar," between each node, each node comprising a minimum of four processors.

The interconnection of the modules uses the memory coherency protocol defined above and makes it possible to perform transactions between modules between two levels of interconnection:

- a first level of interconnection, at the node level, conveying the internal traffic of the node, issuing from the processors and the memories, to the output addresses of the node and of the data ports; and
- a second level of interconnection, at a higher level, transferring the addresses and the data between the various nodes of the system.

The memory access requests (commands and addresses) are broadcast to all of the nodes in order to consult their directories to locate the data that a processor of the system needs to access.

The memory access time is independent of the node in which the physical memory is located.

Two other types of interconnection may be distinguished, according to the type of information processed; interconnection in the address space and interconnection in the data space.

One of the disadvantages of such an architecture resides, in particular, in the utilization of

a "crossbar," which includes a large number of connection points and therefore takes up a lot of space, centralized on the backpanel to which the nodes constituting the machine are connected.

This architecture also requires an "active" backpanel, i.e., one that comprises logical components, as opposed to a "passive" backpanel without any logical components.

5 In this type of machine, it is necessary to duplicate the "crossbar" in order to obtain good availability, and in the event of a failure, part of the usable bandwidth is lost.

A machine of this type is therefore not conducive to the adaptation of new multiprocessor modules to different machine configurations and thus does not allow for the desired modularity.

The object of the invention is specifically to eliminate these drawbacks.

10 One object of the present invention is to propose a modular interconnection architecture that makes it possible to absorb the traffic from all the multiprocessor modules with a variable capacity in terms of the number of processors.

To this end, the first subject of the invention is a modular interconnection architecture for an expandable multiprocessor machine, based on a virtual bus hierarchy, comprising a given  
15 number of multiprocessor modules organized into nodes and distributed on at least two interconnection levels: a first level corresponding to the interconnection of the multiprocessor modules within a node and a second level corresponding to the interconnection of the nodes with one another, the first interconnection level comprising connection agents that connect the multiprocessor modules to one another and handle the transactions between the multiprocessor  
20 modules, the second interconnection level comprising external connection nodes that connect the nodes to one another and handle the transactions between the nodes.

The interconnection architecture according to the invention is characterized in that the connection agents and the external connection nodes, respectively, have the same basic structure, the same external interface, and implement the same coherency control protocol for the cache  
25 memories of the processors.

The second subject of the invention is a process for tracing data blocks in an interconnection architecture as defined above, characterized in that it consists, on the first level, of duplicating in the associative memories only the modified data blocks in the cache memories of the multiprocessor modules, and of tracing only these modified blocks in the node.

The process consists, on the second level, of duplicating in the associative memories of the connection agents of each external connection node only the modified blocks exported, or conversely imported, and of tracing only the modified blocks exported, or conversely imported, between each node of the machine.

5           The third subject of the invention is a process for expanding the capacity of a machine comprising a first given number of first-level processors, organized into a first given number of multiprocessor modules and capable of being inserted into an interconnection architecture as defined above, characterized in that it consists of disconnecting one of the first-level  
10 multiprocessor modules from its connection agent and of connecting, via this freed connection agent, a second given number of processors organized into a second given number of multiprocessor modules, also capable of being inserted into an interconnection architecture as defined above.

15           Finally, the fourth subject of the invention is an expandable multinode multiprocessor machine, characterized in that it comprises an interconnection architecture as defined above.

The invention has the particular advantage of being modular and expandable as desired.

20           In fact, it makes it possible, while still using the same building block, typically a four-processor module associated with its connection agent NCS, or "Network ChipSet," to adapt the number of processors required to cover a very wide range of machines such as UMA, quasi-UMA, or NUMA (Non-UMA) machines, clusters (a term used to designate a set of machines connected by means of a SAN - System Area Network), etc., and to support different operating systems (NT, UNIX, etc.).

NT and UNIX are registered trademarks, and respectively belong to Microsoft and X/Open.

25           The structure of the building block defined above is generally supported by the same silicon ship.

The present invention also eliminates the need for an active "crossbar" in a backpanel (active backpanel).

Other advantages and characteristics of the present invention will emerge through the reading of the following description, given in reference to the attached drawings, which

represent:

- Fig. 1, an exemplary interconnection architecture according to the invention on a first interconnection level MI;
- Fig. 2, the functional diagram of a connection agent NCS according to the invention;
- Fig. 3, an exemplary interconnection architecture according to the invention on a second interconnection level SI;
- Fig. 4, the interconnection architecture of Figs. 1 and 3, developed on both interconnection levels MI and SI;
- Fig. 5, the functional diagram of an external connection node NCE connecting the first-level network MI to the second level network SI according to the invention;
- Fig. 6, an interconnection architecture of the first level MI with five nodes  $N_j$  wherein one of the nodes  $N_j$  is used to connect the second-level network SI;
- Fig. 7, the block diagram of the command and address space of the connection agent NCS of Fig. 2;
- Fig. 8, an interconnection architecture according to the invention in which the first interconnection level SI is reduced to a link.

Fig. 1 schematically illustrates an interconnection architecture for a quasi-UMA machine, typically located on the backpanel of a computer cabinet.

In this example, five multiprocessor modules  $Q_{Pi}$ , with  $i = 0$  to 4, are interconnected by means of an interconnection module MI.

Each multiprocessor module  $Q_{Pi}$  comprises four processors and their respective cache memories, not represented, forming a four-processor module.

The four-processor modules  $Q_{Pi}$  are connected to the interconnection module MI by means of respective network connection agents  $NCS_i$ .

$XI_i$  (eXternal Interface) defines the external interfaces between the four-processor modules  $Q_{Pi}$  and their respective connection agents  $NCS_i$ . They are respectively symbolized by a straight broken line.

The assembly formed by the four-processor module  $Q_{Pi}$ , its connection agent  $NCS_i$  and its external interface  $XI_i$  are considered to constitute a "building block" of the first level of the



interconnection architecture according to the invention.

The interconnection module MI in the example illustrated includes five first-level building blocks QBi, respectively delimited by a closed broken line.

Each building block QBi is linked to the other four via four respective point-to-point links, not referenced in order not to overcomplicate the figure. These links are considered, at the protocol level, as virtual busses. Thus, in the example described, there are four virtual busses and hence four levels of interleaving handled by each connection agent NCSi. Each bus thus allows four accesses, or transactions, in each clock cycle.

The protocol according to the invention makes it possible to control the transactions sent through the interconnection module MI every four clock cycles.

It is based on the following principle:

When a processor sends a request through its bus, for example: read the address of the cache memory containing the data block "a," the only building blocks QBi that can be chosen are those in which there is a cache memory corresponding to this address.

According to the invention, we want to allow the transaction only to the building block QBi that actually contains the block "a" in one of its cache memories, and to consult only the building blocks QBi having this address, without disturbing the others.

For this reason, each connection agent NCSi of the architecture according to the invention, illustrated schematically in Fig. 2, comprises an associative or quasi-associative memory DDi, also known as a "duplicate directory," delimited by a closed broken line.

Each associative memory DDi contains a copy of the contents of the cache memories of the processors of a building block QBi to which it is dedicated.

The contents of the cache memories are organized into four levels of interleaving dd0 through dd3, respectively corresponding to the four processors of the building lock QBi.

Each memory DDi thus contains all the directories of the four processors of the modules QBi, accessible by an address.

The associative memories DDi have a fixed size determined as a function of the number of processors in the building block QBi, and the state of the memories DDi is indicative of the presence of the last data blocks modified in the cache memories of the building block QBi.

Thus, each associative memory DDi plays the role of a filter with respect to the building block QBi to which it is dedicated, in the sense that a transaction is only carried out when an associative memory DDi contains the last modified copy of the block "a".

The dotted lines represent the path, or "snooping," followed by the protocol in order to consult the state of the associative memories DDi before beginning a transaction.

The connection agent NCSi comprises, on the right side of the figure, the external interface XI for connecting to a four-processor module QPi.

It is connected, on the left side of the figure, via the four inbound and outbound links of Fig. 1, to the other four building blocks QBi of the interconnection module MI.

The protocol according to the invention is based on the principle according to which only the modified blocks are duplicated in the associative memories DDi of the connection agents NCSi.

Thus, it is possible, with an architecture on a level according to that of Fig. 1 and connection agents NCS according to Fig. 2, to interconnect, and to handle the flow of transactions typically exchanged between, 4 to 20 processors.

Hereinafter, this architecture will be considered to define a first level MI of a modular architecture according to the invention.

Fig. 3 illustrates, in a diagram similar to that of Fig. 1, a second level of the modular architecture defined above that integrates the first level MI, which can be considered to be a building block for the second level SI.

According to this embodiment, five second-level building blocks HBj, with  $j = 0$  to 4, are connected to the interconnection system SI via their respective connection agents NCS"j and their respective interfaces XI"j. The interconnection system SI, with respect to the blocks HBj of the second level, behaves like the interconnection module MI with respect to the blocks QBi of the first level.

Fig. 4 schematically illustrates an architecture similar to the one illustrated in Fig. 3, developed on both interconnection levels MI and SI. It typically makes it possible to interconnect up to 80 processors for second-level building blocks HBj, respectively comprising 16 processors. Each second level building block HBj respectively includes four first-level building blocks QBi,

which themselves respectively include 4 processors.

Expanding the capacity of a machine consists of increasing its number of processors.

The expansion process according to the invention consists, in a first step, of disconnecting one of the five four-processor modules QPi of the first level, the module QP3 in the example in question, from its connection agent NCS3, and in a second step, of connecting the second-level architecture SI using the freed connection agent, NCS3 in the example illustrated in Fig. 1, which will be renamed NCS'j, with  $j = 0$  in the example in question.

The interconnection system SI is connected to the interconnection module MI via a connection agent NCS'0 disposed head-to-tail relative to the agent NCS'0, the external interface XI'0 being physically disposed between the first and the second connection agents NCS'0 and NCS'0.

The two connection agents NCS'j and NCS''j, connected head-to-tail, illustrated schematically in Fig. 5, are therefore seen by the interconnection system SI as a single connection agent NCS''j, as represented in Fig. 3. The two connection agents NCS'j and NCS''j, connected head-to-tail, and their interface XI'j, define an external connection node NCEj through which pass the transactions between the various processors on the first level MI and those of the second level SI in accordance with the protocol according to the invention.

The protocol according to the invention makes it possible to maintain the coherency of the various cache memories distributed on the two interconnection levels MI and SI by consulting the respective associative memories of the agents NCS'j and NCS''j. One of the characteristics of the protocol according to the invention is its tracing capacity, not only for a building block QBi of the first level MI, but for all the building blocks QBi and HBj of both levels MI and SI.

In order to absorb the traffic between the two levels MI and SI while retaining the same bandwidth XIi and XI''j and the same building block for both levels MI and SI, i.e., associative memories DDi, DDi' and DD''j of the same size no matter what the number of levels, the latency of the second level SI is chosen so as to be greater than that of the first level MI. It is typically equal to one clock cycle for the first level MI and equal to two clock cycles for the second level SI. It is thus possible to construct, from a modular interconnection architecture according to the

invention and consistent with the examples illustrated in Figs. 1 through 5, at least two types of machines: a first machine with 4 to 20 processors and a second machine with more than 60 processors.

Two distinct modes are defined for parameterizing an external connection node NCEj as a function of the level of the interconnection architecture in which the desired information is located: the MI mode and the SI mode.

According to the trace principle established in the MI mode, only the modified blocks are duplicated. In the SI mode, the trace is only performed on the modified blocks exported. The exportation of modified blocks is understood to go in the direction from the first level MI to the second level SI, or from the lower level to the higher level. Symmetrically, the importation of modified blocks is understood to go in the direction from the second level SI to the first level MI, or from the higher level to the lower level.

When a processor of a first node N0 requests to exclusively read (read-invalidate "a") a block "a" located in another node Nj, the request is broadcast to all of the nodes Nj via the four virtual busses of the interconnection system SI.

This broadcasting may be likened to the known broadcast mode that applies to a multiple bus; the ten point-to-point links of the interconnection system SI behave like four virtual busses in the protocol sense.

The external connection node NCEj behaves like an external agent when it is parameterized in the external mode.

In this (multinode) configuration, the acknowledgement of an invalidate command is the aggregate, in the external agent of the requesting module, of the partial acknowledgements sent by the external agents of the other nodes Nj.

The agent NCS" has the same function, for the second level (16 processors) that the agent NCS has for the first level (4 processors).

The main operational differences between the agent NCSi and the agent NCS"j are summarized in the following table with respect to the aspects of link latency, associative memory, invalidate commands, and read-and-invalidate commands:

|                      | NCSi (MI)  | NCS"J (SI)  |
|----------------------|--|---|
| latency of the links | 1 clock cycle<br>(backpanel)   | 2 clock cycles<br>(cables)  |
| associative memory   | duplication of the 4 cache memories of the processors                          | partial duplication of the 16 cache memories of the processors in the node (only the imported modified blocks are traced with precision)                        |
| invalidate           | filtered propagation to the cache memories of the 4 processors in the module   | propagation to the NCSi of each of the 4 building blocks of the module (explicit acknowledgement by means of a response through the network from the module MI) |
| read and invalidate  | the supplying of the data is contingent on acknowledgement of the invalidation | its acknowledgement requires acknowledgement from the nodes of the system; no blocking of the data  |

Likewise, the table below summarizes the operational differences between an agent NCSi and an agent NCS'j with respect to the aspects of associative memory, invalidate commands, and read-and-invalidate commands.

5 The agent NCS'j is disposed in the same place as an agent NCSi and handles traffic issuing from 64 processors rather than the 4 processors handled by the agent NCSi.

The architecture considered in this table is illustrated in Fig. 6.

|                        | NCS <sub>i</sub>   | NCS <sub>j</sub>   |
|------------------------|--|--|
| associative memory     | duplication of the 4 cache memories of the processors                          | partial duplication of the 64 cache memories (only the imported and modified blocks are traced with precision) |
| invalidate             | filtered propagation to the cache memories of the 4 processors in the node     | propagation to the 64 processors on the second level (explicit notification)                                   |
| reading and invalidate | the supplying of the data is contingent on acknowledgement of the invalidation | its acknowledgement requires acknowledgement from the nodes of the system; no blocking of the data             |

The external interfaces XI<sub>i</sub> and XI<sub>j</sub> have similar bandwidths.

When a requested block is identified in one of the associative memories DD<sub>i</sub> of the building blocks QBi, the requested block is "exported" from the cache memory containing the last updated copy of the block to the requesting processor. The new valid copy is then located in the cache memory of the requesting processor.

It is the coherency control protocol according to the invention that first searches for the associative memory DD<sub>i</sub> in which the modified block is located, then secondly, searches for the associative memory DD<sub>j</sub> and DD<sub>j</sub> of the external connection node NCE<sub>j</sub> that contains the information: the modified and exported block "a". The building block QBi containing the requesting processor is also identified in the node N<sub>j</sub>.

The principle is symmetrical for the imported blocks. With this trace principle, based on the consultation of the associative memories of the connection agents NCS<sub>i</sub> and the external

connection nodes  $NCE_j$ , the bandwidth can remain constant for both levels and absorb disparate traffic between the various first- and second-level building blocks  $QBi$  and  $HBj$ .

The memory coherency control protocol according to the invention is thus capable of enabling the architecture to absorb transaction traffic in the worst case scenario, in which four of the five second-level building blocks  $HBj$  would simultaneously search for a block in the fifth building block.

Fig. 7 illustrates the block diagram of a connection agent  $NCSi$  according to the invention, considered in the command and address space.

The description of this diagram will make it possible to better understand the coherency control protocol used by the connection agent  $NCSi$  according to the invention.

All of the primary functions implemented on the first and second levels  $MI$  and  $SI$  are represented.

To simplify the figure, only the blocks that relate to the first level ( $i=0$ ) of the four levels of interlacing are represented.

A first block  $RL$  performs a logical sequencing of the external requests received at the input of the agent (on the left side of the figure). The requests are sequenced in the same way no matter what the sending building block. For each of the inputs, the block  $RL$  has the capacity to handle several requests received simultaneously, in order to prevent a loss of requests in case of an unexpected halt in the flow of requests.

Logical filters  $RFi$  respectively receive the requests in the order imposed by the block  $RL$ . They respectively filter all the requests assigned to the same level of interlacing  $i$  and to the same system identifier of the building block.

Arbitration blocks  $ARBi$  are respectively connected to the outputs of the filters  $RFi$ . They select a request in accordance with a given algorithm, for example an equitable algorithm of the "Round Robin" type, wherein the last request chosen becomes the one with the lowest priority.

Associative memories  $DDi$  are respectively connected to the outputs of the arbitration blocks  $ARBi$ . Each associative memory  $DDi$  contains a copy  $ddij$  of the content of the cache memory of each processor of a building block. In the  $SI$  mode, the associative memories  $DD'j$  and  $DD''j$  respectively contain either a copy of the modified blocks exported or a copy of the

modified blocks imported, depending on the direction of the transaction between the two levels MI and SI.

To force an input, i.e., to impose a new datum on an input, an invalidate command or a read and invalidate command is necessary in order to be able to reuse a cache memory that has already been validated in the associative memory DDi.

To do this, extensions DDEAMi to the associative memories DDi are connected to their outputs and are for slowing down the transactions intended to pass through the bus while waiting for the bus to be available (idle).

All the requests are serialized on output from the arbitration blocks ARBi and are then stored in a buffer memory SERB in which they are organized according to their status (internal/external, home/remote, hit/miss, shared/exclusive/modified) into four queues Fi, respectively corresponding to the four levels of interlacing  $i = 0$  to 3. Thus, four requests can be accepted in each clock cycle.

A multiplexer MUX1, on the right side of the figure, receives the external requests from the four queues Fi, and after validation by the memory extensions DDEAMi, the requests are sent, one by one, from the output of the connection agent NCS, which also corresponds to the output of the multiplexer MUX1, to the bus of the destination building block.

Blocks UAMi handle the updating of the associative memories DDi. There is one block per associative memory. These blocks UAMi store the updates that must be performed in the associative memories DDi, and they are executed when there is no traffic, or simultaneously, by all the building blocks.

Storage blocks IRIPi store all the transactions that are addressed to the same building block and that must wait for an end-of-transaction event such as:

- end of access to main storage (reading of a transaction S, I);
- update of the main storage (reading of a partial write transaction M, E). In the case of E, the data may be invalidated, thus indicating that the potential holder no longer holds the data and that the "home," i.e., the local cache memory containing the data block, must supply it;
- end of transmission of an invalidate command;



and the transactions that have ended in a "hit E" or "hit M" in the building block, i.e, an access that has resulted in the state E or an access that has resulted in the state M, and that must wait for a given event to terminate them: end of read and invalidate in the system bus of the building block.

5 A logical block IA receives the responses related to the internal requests, issued by the arbitration blocks ARBi, and indicates for each of them whether they have succeeded "hit" or failed "miss" (defer: wait for its turn or retry because of a conflict).

A block ESA receives the responses to the consultations of the associative memories DDi and broadcasts these results to the other building blocks.

10 A block SAH receives and analyzes the responses to consultations of the following types:

- a consultation of the read type (Read):
  - validates or clears an anticipated read request and waits for an update of the block (MI mode);
  - prepares a response to the sender of the request, and to the processor holding the requested block if it is different (SI mode);
- a consultation of the partial write type (PartialWrite):
  - prepares the update of the block of the holding processor: writing into the block (SI mode).

15 A block HVB receives information on the filled state of certain storage blocks such as the blocks UAMi, DDEAMi, and informs the other building blocks in case of an imminent overflow of one of these blocks, so that they stop sending requests addressed to these blocks.

20 A partial write request (PartialWrite) is seen by a remote building block as a read-and-invalidate request.

25 A write generating block WG collects the requests sent through the bus from the output of the connection agent, which also corresponds to the output of the multiplexer MUX1, and updates the local building block by merging the data that has been partially written and the data that has been read.

Blocks ORIPi store all the operations requested by a given building block (read-wait for data request, invalidate-wait for acknowledgement request, etc.). These blocks examine the

requests addressed to each node  $N_j$  issuing from a building block, not represented but which would be on the right side of the figure, in order to detect a collision between the current outbound local requests and inbound remote requests.

The respective outputs of the blocks  $ORIP_i$  and  $IRIP_i$  are multiplexed by multiplexers  $MUX_i$  whose outputs are respectively connected to the inputs of the blocks  $UAM_i$ . The updates of the memories  $DD_i$  are performed either in a transaction resulting from external requests via the blocks  $IRIP_i$  that therefore return to the connection agent NCS (top left of the figure), for example when a processor of a given node  $N_j$  wants to read a given block, or in a transaction resulting from internal requests via the block  $ORIP_i$  that leave the connection agent NCS (bottom left of the figure), for example when a block is "kicked out" of the cache of a cache memory of a given processor.

A block CM collects the system identifier of the building block in which the desired storage space is located (the cache memory of the processor hosting the block that is the object of the request is there!).

A block SAH receives in a first series of inputs the responses to the consultations of the other building blocks. It performs a synthesis of these responses, which makes it possible to validate or invalidate the anticipated external read operations and also to know whether a request will be honored internally or externally.

In fact, the inputs of the local block SAH correspond to the outputs of the blocks ESA of the other building blocks.

A multiplexer  $MUX_3$ , at the bottom left of the figure, receives the internal requests and the block writing operations linked to partial write operations issued by the block WG. It makes it possible to serialize the requests, which are then broadcast to the other four building blocks, respectively through the four links linking the sending building block to the other blocks, respectively via four output registers  $REG_0$  through  $REG_3$ .

A fifth register  $REG_{in}$ , at the right of the figure, is the input register of the agent NCS. It receives the internal requests arriving in the building block. It is controlled by the information issued by the logical block IA.

The same connection agent NCS, a functional description of which is given above, can be

used in three different applications:

- for connecting a four-processor module QPi to the module MI (Fig. 1);
- for connecting a node Nj to the system SI (Figs. 3, 4 and 6); and
- for connecting two nodes Nj (Fig. 8).

These three applications are implemented by the same component (chipset).

The processing and the path of this data through the connection agent NCS are not represented.

They are limited to the routing of the data, either from the interface XI to the module MI or the system SI or in the opposite direction, to the multiplexing of the data issuing from the module MI or the system SI.

The processing of the data is controlled by the address space of the connection agent NCS. This space supervises all the inbound and outbound data flows and if necessary, performs a translation between external or internal data labels.

The interface XI is a de-multiplexed interface and therefore comprises at least a command/address part and a data part.

To prevent a read response from overlapping an invalidate command, the read response must wait until all the commands serialized prior to the read request have been sent through the system bus of the four-processor module.

Fig. 8 illustrates a variant of embodiment of an architecture according to the invention.

This variant corresponds to an intermediate configuration limited to interconnecting between two nodes Nj with five four-processor modules QPi each.

In this configuration, in one module QPi of each node Nj, only the connection agent NCS' is retained, and the two nodes Nj are connected to one another via their respective connections agents NCS'.

This results in a 32-processor machine with a second interconnection level that is reduced to a link.

Thus, an interconnection without a system network SI is possible using only two connection agents NCS' connected head-to-tail, with an interface of the XI' type as the only external interface.

TYSO01 9107991v0|T2147-906398|03\31\00

## CLAIMS

1           1.     Modular interconnection architecture for an expandable multiprocessor machine,  
2     based on a virtual bus hierarchy, comprising a given number of multiprocessor modules (QPi)  
3     organized into nodes (Nj) and distributed on at least two interconnection levels: a first level (MI)  
4     corresponding to the interconnection of the multiprocessor modules (QPi) within a node (Nj),  
5     and a second level (SI) corresponding to the interconnection of the nodes (Nj) with one another,  
6     the first interconnection level (MI) comprising connection agents (NCSi) that connect the  
7     multiprocessor modules (QPi) to one another and handle the transactions between the  
8     multiprocessor modules (QPi), the second interconnection level (SI) comprising external  
9     connection nodes (NCE) that connect the nodes (Nj) to one another and handle the transactions  
10    between the nodes (Nj), characterized in that the connection agents (NCSi) and the external  
11    connection nodes (NCEj) respectively have the same basic structure, the same external interface  
12    (XI), and implement the same coherency control protocol for the cache memories of the  
13    processors.

14           2.     Architecture according to claim 1, characterized in that each external connection  
15    node (NCEj) comprises two identical connection agents (NCSi) connected head-to-tail, one of  
16    the two agents (NCS'j) receiving and filtering the transactions sent by the node (Nj) to which it is  
17    connected, and the other (NCS''j) receiving and filtering the transactions sent by the other nodes  
18    (Nj) to which it is connected.

19           3.     Architecture according to claim 1, characterized in that each connection agent  
20    (NCSi) comprises an associative memory (DDi) with a fixed size determined as a function of the  
21    number of processors in the multiprocessor module (QPi) to which the connection agent (NCSi)  
22    is connected, the state of the memories (DDi) being indicative of the presence of the last  
23    modified data blocks in the cache memories of the multiprocessor module (QPi).

24           4.     Architecture according to claim 2, characterized in that the connection agents

(NCS'<sub>j</sub> and NCS''<sub>j</sub>) respectively comprise an associative memory (DD'<sub>j</sub> and DD''<sub>j</sub>) with a fixed size determined as a function of the number of processors in the multiprocessor modules (HB<sub>j</sub>) to which they are connected, the state of the associative memories (DD'<sub>j</sub> and DD''<sub>j</sub>) being indicative of the presence of the last modified data blocks exported, or conversely imported.

5. Architecture according to claim 2, characterized in that the first and second head-to-tail connection agents (NCS'<sub>j</sub> and NCS''<sub>j</sub>) only accept transactions for the blocks that are modified in their respective associative memories (DD'<sub>j</sub> and DD''<sub>j</sub>); the modified data blocks in the first connection agent (NCS'<sub>j</sub>) are exported to the requesting multiprocessor module or modules and, conversely, the modified data blocks in the second connection agent (NCS''<sub>j</sub>) are imported from the module or modules holding the blocks.

6. Architecture according to any of claims 1 through 5, characterized in that the latency of the second interconnection level (SI) is double the latency of the first interconnection level (MI).

7. Process for tracing data blocks in an interconnection architecture according to any of claims 1 through 6, characterized in that it consists, on the first level, of duplicating in the associative memories (DD<sub>i</sub>) only the modified data blocks in the cache memories of the multiprocessor modules (QP<sub>i</sub>) and of tracing only the modified blocks inside the node (N<sub>j</sub>).

8. Process for tracing data blocks in an interconnection architecture according to any of claims 1 through 6, characterized in that it consists, on the second level (SI), of duplicating in the associative memories (DD'<sub>j</sub> and DD''<sub>j</sub>) of the connection agents (NCS' and NCS''<sub>j</sub>) of each external connection node (NCE<sub>j</sub>) only the modified blocks exported, or conversely imported, and of tracing only the modified blocks exported, or conversely imported, between each node (N<sub>j</sub>) of the machine.

9. Process for expanding the capacity of a machine comprising a first given number

2 of processors on a first level (MI) organized into a first given number of multiprocessor modules  
 3 (QPi) and capable of being inserted into an interconnection architecture according to any of  
 4 claims 1 through 6, characterized in that it consists of disconnecting one of the first-level  
 5 multiprocessor modules (QPi) from its connection agent (NCSi) and of connecting, via this freed  
 6 connection agent, a second given number of processors organized into a second given number of  
 7 multiprocessor modules, also capable of being inserted into an interconnection architecture  
 8 according to any of claims 1 through 6.

1 10. Process according to claim 9, characterized in that, the second given number of  
 2 processors being organized into a second given number of multiprocessor modules on the second  
 3 level (SI), it consists of connecting it to the connection agent (NCSi) of the first given number of  
 4 processors on the first level (MI) through one of the connection agents (NCS"j) on the second  
 5 level.

6 11. Process according to claim 10, characterized in that, the second given number of  
 7 processors also being on the first level (MI), it consists of connecting the respective connection  
 8 agents (NCS') to the first and second given numbers of processors, the second level (SI) being  
 9 reduced to a single link.

10 12. Expandable multinode multiprocessor machine, characterized in that it comprises  
 11 an interconnection architecture according to any of claims 1 through 6.

**ABSTRACT**

The present invention relates to a modular interconnection architecture for an expandable multiprocessor machine. It comprises a first interconnection level (MI) comprising connection agents (NCS<sub>i</sub>) that connect the multiprocessor modules and handle the transactions between the multiprocessor modules, and a second interconnection level (SI) comprising external connection nodes (NCE<sub>j</sub>) that connect the nodes (N<sub>j</sub>) to one another and handle the transactions between the nodes (N<sub>j</sub>). Each external connection node (NCE<sub>j</sub>) comprises two connection agents identical to the connection agent (NCS<sub>i</sub>), connected head-to-tail, one of the two agents (NCS'<sub>j</sub>) receiving and filtering the transactions sent by the node (N<sub>j</sub>) to which it is connected, and the other agent (NCS''<sub>j</sub>) receiving and filtering the transactions sent by the other nodes (N<sub>j</sub>) to which it is connected.

Its applications specifically include the construction of an entire range of machines: UMA, QUASI-UMA, NUMA, cluster, etc.

FIG. 5



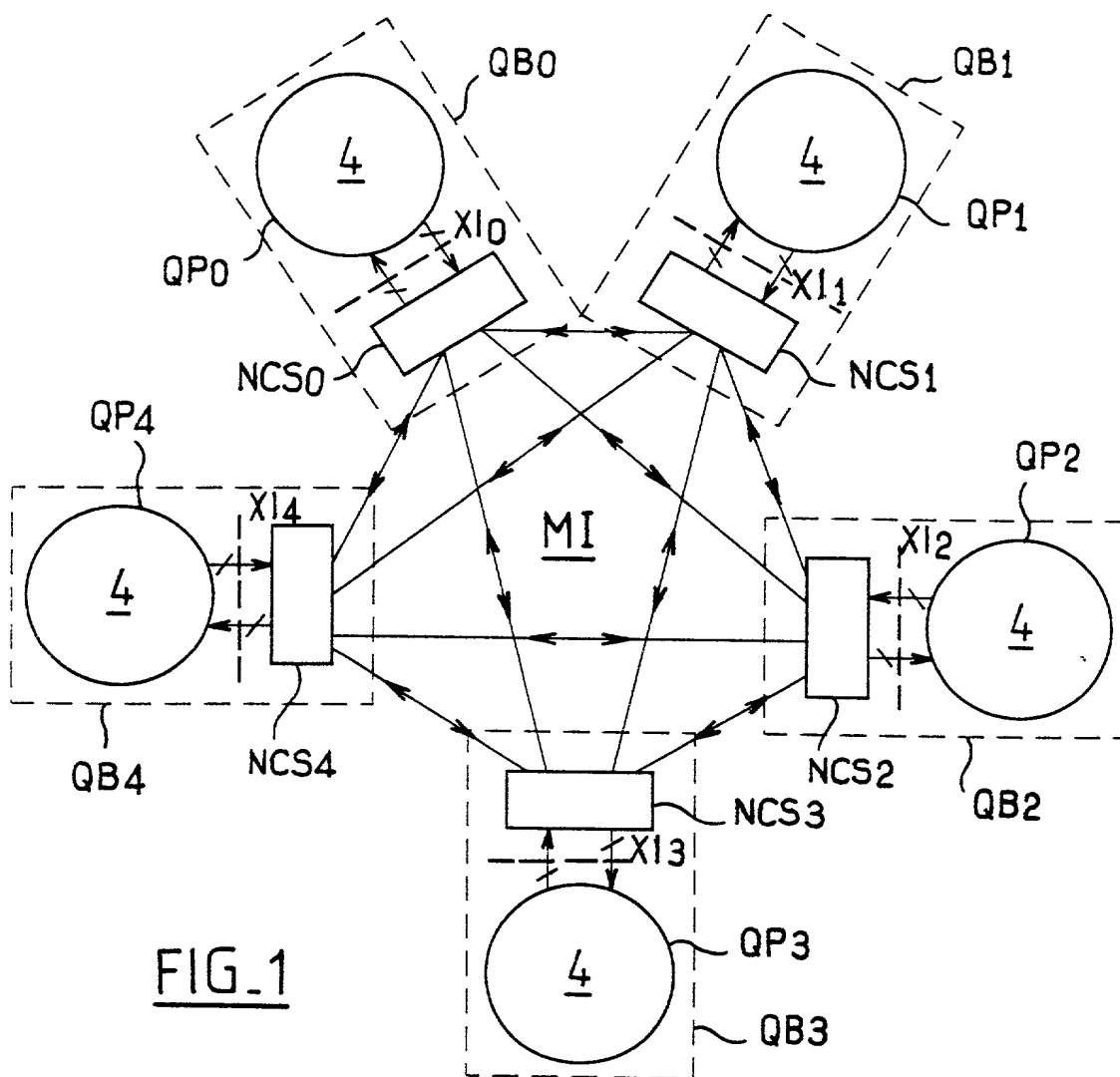


FIG. 1

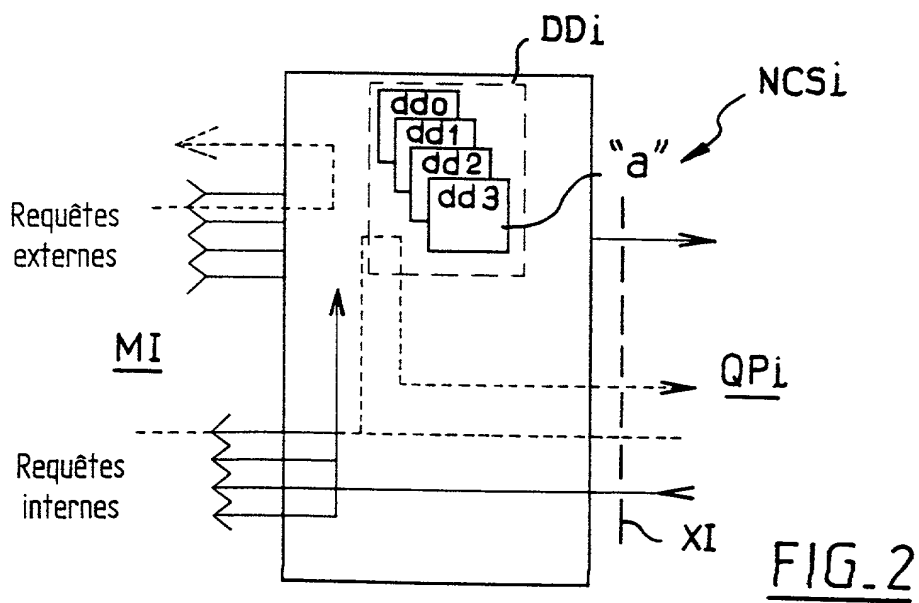
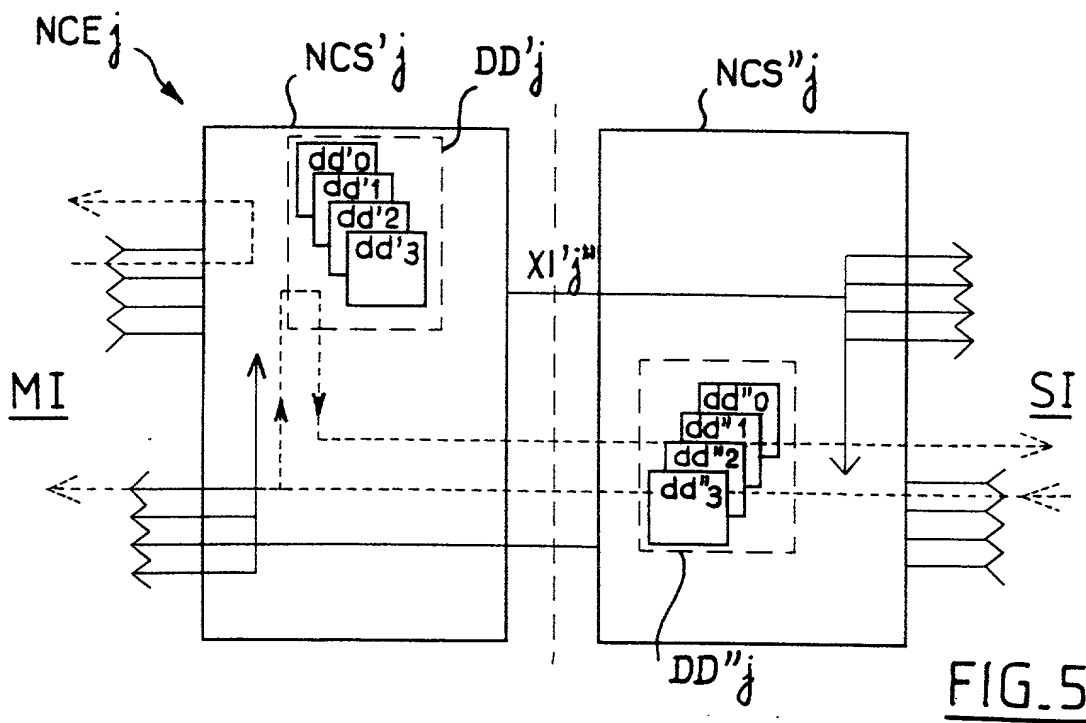
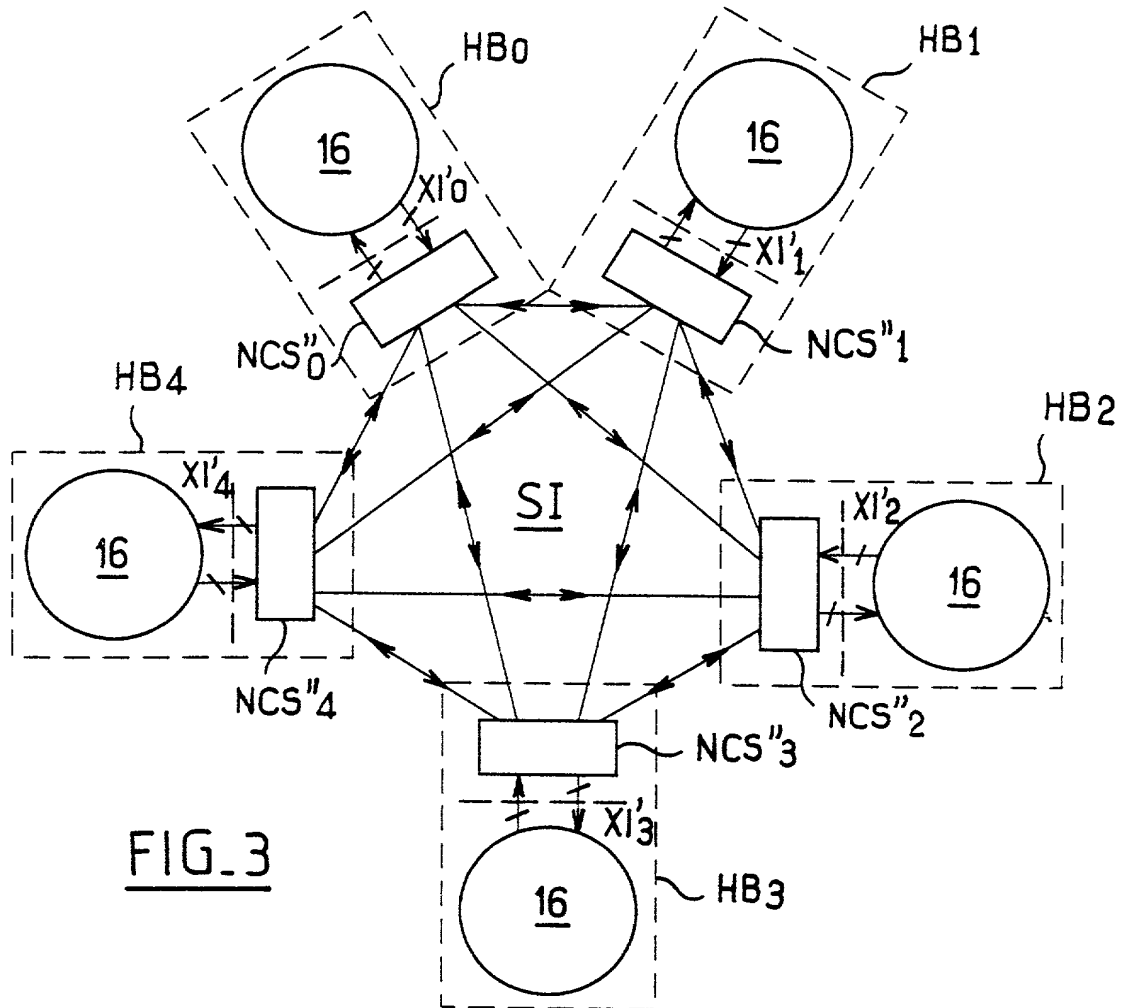


FIG. 2



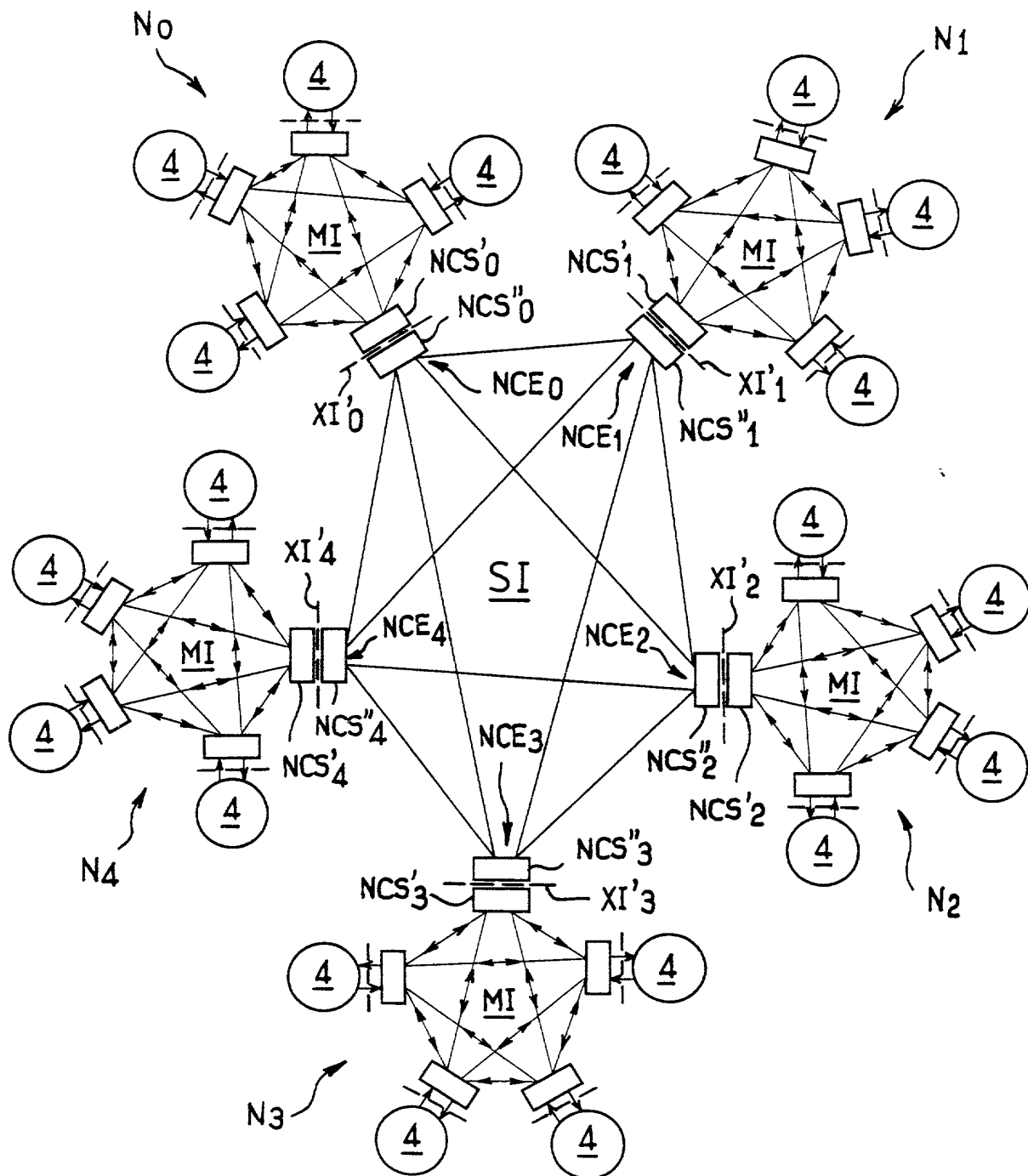


FIG. 4

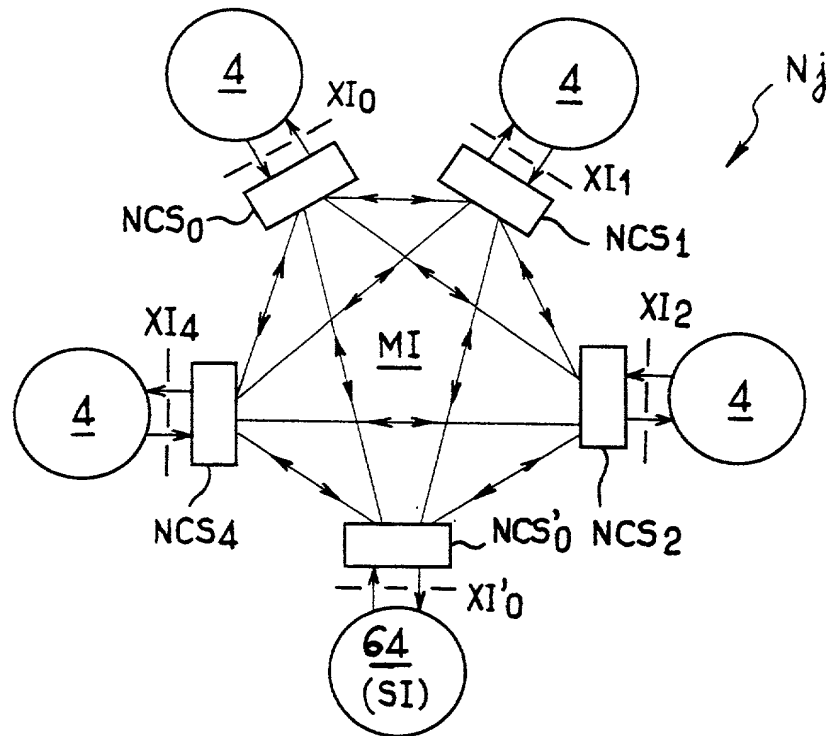


FIG. 6

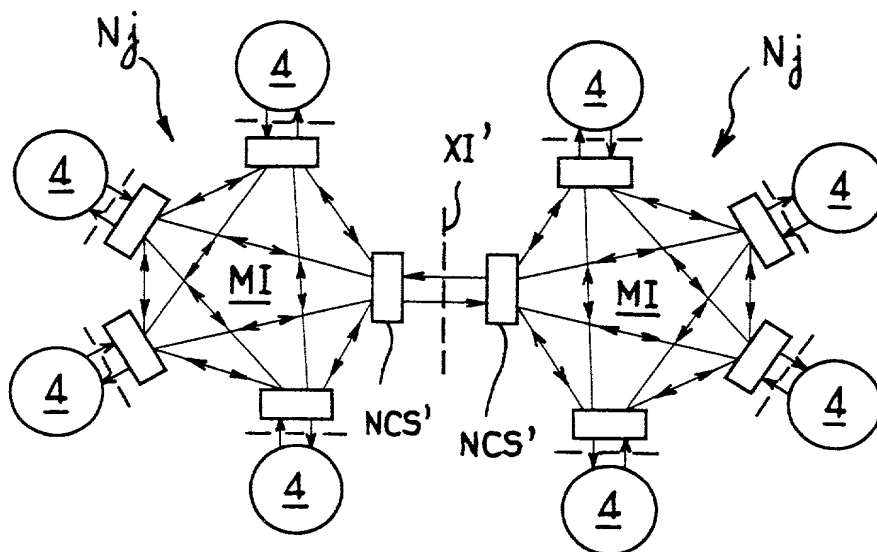
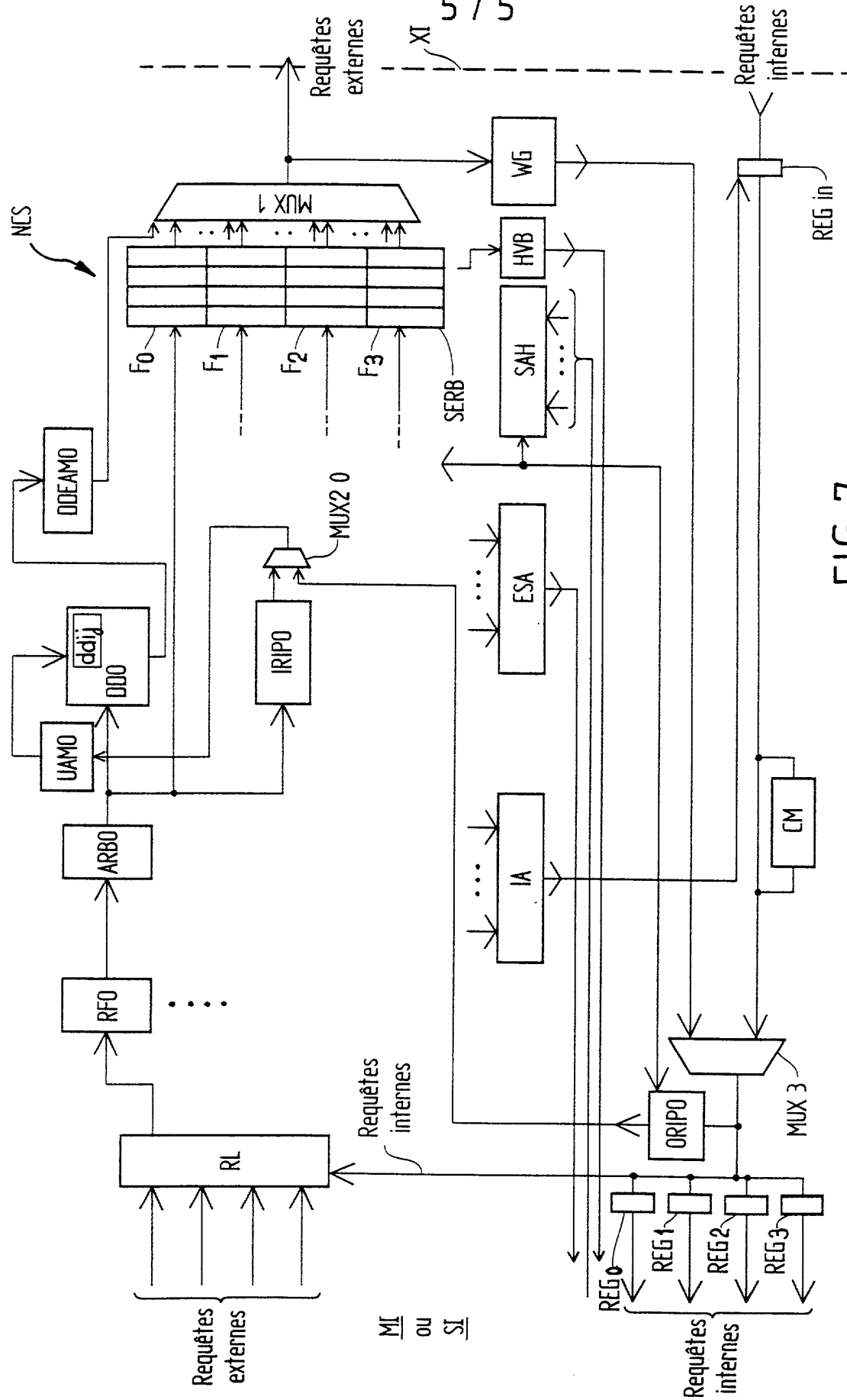
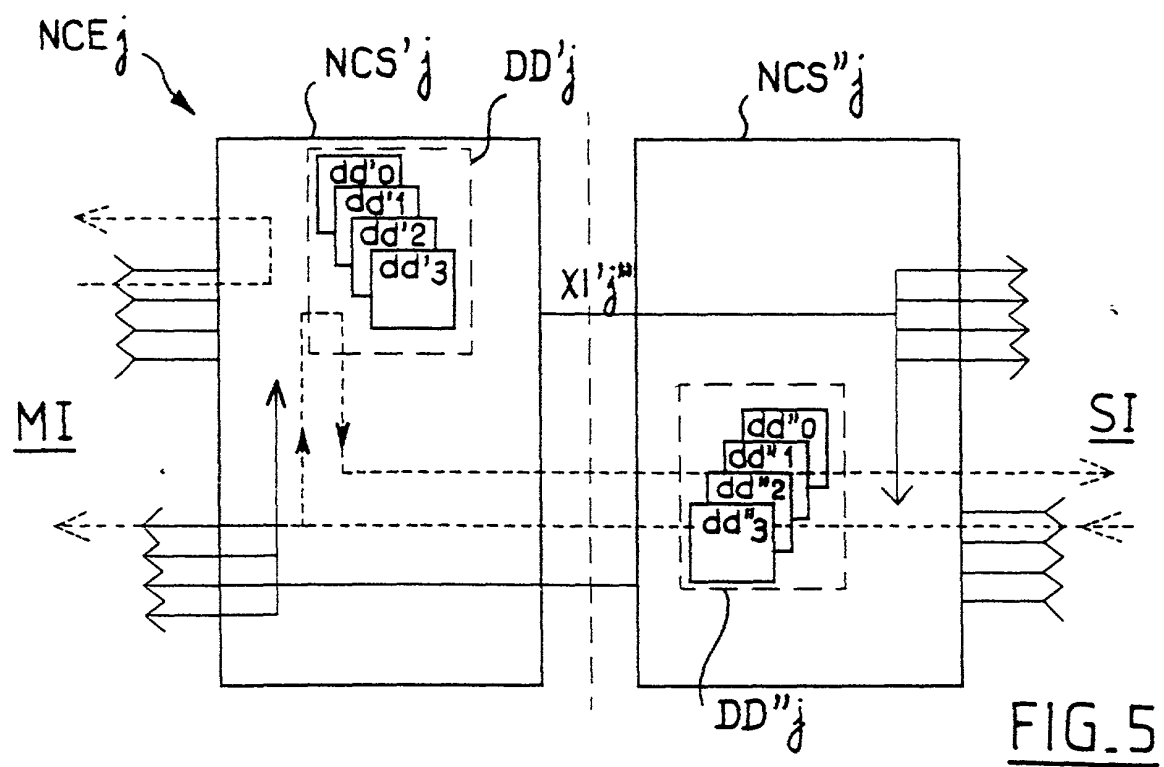


FIG. 8



**FIG. 7**

Figure de l'abrégé



# Declaration and Power of Attorney for Patent Application

## Déclaration et Pouvoirs pour Demande de Brevet

### French Language Declaration

En tant que l'inventeur nommé ci-après, je déclare par le présent acte que:

Mon domicile, mon adresse postale et ma nationalité sont ceux figurant ci-dessous à côté de mon nom.

Je crois être le premier inventeur original et unique (si un seul nom est mentionné ci-dessous), ou l'un des premiers co-inventeurs originaux (si plusieurs noms sont mentionnés ci-dessous) de l'objet revendiqué, pour lequel une demande de brevet a été déposée concernant l'invention intitulée

**Architecture d'interconnexion modulaire pour machine multiprocesseur extensible, mettant en œuvre une hiérarchie de bus virtuelle à plusieurs niveaux et la même brique de base pour tous les niveaux.**

et dont la description est fournie ci-joint à moins que la case suivante n'ait été cochée:

- ☐ a été déposée le \_\_\_\_\_  
sous le numéro de demande des Etats-Unis ou le  
numéro de demande international PCT  
\_\_\_\_\_ et modifiée le  
\_\_\_\_\_ (le cas échéant).

Je déclare par le présent acte avoir passé en revue et compris le contenu de la description ci-dessus, revendications comprises, telles que modifiées par toute modification dont il aura été fait référence ci-dessus.

Je reconnais devoir divulguer toute information pertinente à la brevetabilité, comme défini dans le Titre 37, § 1.56 du Code fédéral des réglementations.

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

\_\_\_\_\_

\_\_\_\_\_

the specification of which is attached hereto unless the following box is checked:

- ☐ was filed on \_\_\_\_\_  
as United States Application Number or PCT  
International Application Number  
\_\_\_\_\_ and was amended on  
\_\_\_\_\_ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56.

**French Language Declaration**

Je revendique par le présent acte avoir la priorité étrangère, en vertu du Titre 35, § 119(a)-(d) ou § 365(b) du Code des Etats-Unis, sur toute demande étrangère de brevet ou certificat d'inventeur ou, en vertu du Titre 35, § 365(a) du même Code, sur toute demande internationale PCT désignant au moins un pays autre que les Etats-Unis et figurant ci-dessous et, en cochant la case, j'ai aussi indiqué ci-dessous toute demande étrangère de brevet, tout certificat d'inventeur ou toute demande internationale PCT ayant une date de dépôt précédant celle de la demande à propos de laquelle une priorité est revendiquée.

Prior foreign application(s)

Demande(s) de brevet antérieure(s)

99 05266 France  
(Number) (Country)  
(Numéro) (Pays)

(Number) (Country)  
(Numéro) (Pays)

Je revendique par le présent acte tout bénéfice, en vertu du Titre 35, § 119(e) du Code des Etats-Unis, de toute demande de brevet provisoire effectuée aux Etats-Unis et figurant ci-dessous.

(Application No.) (Filing Date)  
(N° de demande) (Date de dépôt)

(Application No.) (Filing Date)  
(N° de demande) (Date de dépôt)

Je revendique par le présent acte tout bénéfice, en vertu du Titre 35, § 120 du Code des Etats-Unis, de toute demande de brevet effectuée aux Etats-Unis, ou en vertu du Titre 35, § 365(c) du même Code, de toute demande internationale PCT désignant les Etats-Unis et figurant ci-dessous et, dans la mesure où l'objet de chacune des revendications de cette demande de brevet n'est pas divulgué dans la demande antérieure américaine ou internationale PCT, en vertu des dispositions du premier paragraphe du Titre 35, § 112 du Code des Etats-Unis, je reconnais devoir divulguer toute information pertinente à la brevetabilité, comme défini dans le Titre 37, § 1.56 du Code fédéral des réglementations, dont j'ai pu disposer entre la date de dépôt de la demande antérieure et la date de dépôt de la demande nationale ou internationale PCT de la présente demande:

(Application No.) (Filing Date)  
(N° de demande) (Date de dépôt)

(Application No.) (Filing Date)  
(N° de demande) (Date de dépôt)

Je déclare par le présent acte que toute déclaration ci-incluse est, à ma connaissance, véridique et que toute déclaration formulée à partir de renseignements ou de suppositions est tenue pour véridique; et de plus, que toutes ces déclarations ont été formulées en sachant que toute fausse déclaration volontaire ou son équivalent est passible d'une amende ou d'une incarcération, ou des deux, en vertu de la Section 1001 du Titre 18 du Code des Etats-Unis, et que de telles déclarations volontairement fausses risquent de compromettre la validité de la demande de brevet ou du brevet délivré à partir de celle-ci.

I hereby claim foreign priority under Title 35, United States Code, § 119(a)-(d) or § 365 (b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT International application which designated at least one country other than the United States, listed below, and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

26.04.1999  
(Day/Month/Year Filed)  
(Jour/Mois/Année de dépôt)

(Day/Month/Year Filed)  
(Jour/Mois/Année de dépôt)

Priority Claimed  
Droit de priorité revendiqué

☒☐

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below.

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s), or § 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application.

(Status) (patented, pending, abandoned)  
(Statut) (breveté, en cours d'examen, abandonné)

(Status) (patented, pending, abandoned)  
(Statut) (breveté, en cours d'examen, abandonné)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.



Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**French Language Declaration**

POUVOIRS: En tant que l'inventeur cité, je désigne par la présente l'(les) avocat(s) et/ou agent(s) suivant(s) pour qu'ils poursuive(nt) la procédure de cette demande de brevet et traite(nt) toute affaire s'y rapportant avec l'Office des brevets et des marques: (mentionner le nom et le numéro d'enregistrement).

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith: (list name and registration number)

Edward J. Kondracki, Reg. No. 20,604

Dennis P. Clarke, Reg. No. 22,549

William L. Feeney, Reg. No. 29,918

John C. Kerins, Reg. No. 32,421

Send Correspondence to:

Miles &amp; Stockbridge P.C.

1751 Pinnacle Drive, Suite 500

McLean, VA 22102-3833

Direct Telephone Calls to:

(name and telephone number)

703-903-9000

Adresser toute correspondance à:

Adresser tout appel téléphonique à:  
(nom et numéro de téléphone)

|  |  |
|--|--|
| Nom complet de l'unique ou premier inventeur<br><b>Autechaud Jean-François</b> | Full name of sole or first inventor        |
| Signature de l'inventeur <i>JF. Autechaud</i> Date <i>6 mars 2000</i>          | Inventor's signature Date                  |
| Domicile <b>29, rue des Archives<br/>75004 Paris - France</b>                  | Residence                                  |
| Nationalité <b>Française</b>   | Citizenship                                |
| Adresse postale <b>29, rue des Archives<br/>75004 Paris - France</b>           | Post Office Address                        |
| Nom complet du second co-inventeur, le cas échéant                             | Full name of second joint inventor, if any |
| Signature du second inventeur Date   | Second Inventor's signature Date           |
| Domicile   | Residence                                  |
| Nationalité  | Citizenship                                |
| Adresse postale  | Post Office Address                        |
|  |  |

(Fournir les mêmes renseignements et la signature de tout co-inventeur supplémentaire.)

(Supply similar information and signature for third and subsequent joint inventors.)